Test Strategy for To-Do List Web Application

Contents

1 Introduction

The Test Strategy communicates the testing approach for the To-Do List Web Application, covering manual testing and automation scripts. It outlines objectives, scope, roles, assumptions, constraints, test approach, environment, schedule, deliverables, test cases, and bug reporting, ensuring alignment with future automation efforts.

1.1 Objectives

The primary objective is to verify that the To-Do List Web Application meets all functional, usability, and boundary requirements, ensuring users can manage tasks effectively. The secondary objective is to identify and document defects, communicating them to the development team for resolution before release.

1.2 Roles and Responsibilities

Role	Responsibilities
Test Lead	Develop Test Plan, manage defect reporting, conduct bug triage, produce reports.
Tester	Create & Execute manual test cases, log defects, verify bug fixes.
Developer	Develop application, fix defects, support test environment setup.
Project Manager	Oversee timelines, review deliverables, approve test completion.

2 Scope

2.1 In Scope

- Functional testing: Add, edit, delete, mark tasks complete/incomplete, view tasks, login/logout.
- UI testing: Button visibility, form alignment, error messages, responsiveness.
- Boundary testing: Max title length, empty inputs, max tasks, special characters.
- Manual test case execution, designed for future automation.
- Automation scripts for opening app, adding task, marking complete, deleting task.

2.2 Out of Scope

- Performance, security, and mobile-specific testing.
- Migration of existing test cases.

3 Assumptions

- Application is accessible in a test environment with Chrome (latest version).
- Testers are trained on manual testing and defect reporting.
- Requirements are finalized before testing.

4 Constraints

- Testing is primarily manual in Phase 1, with automation planned for future phases.
- Test environment setup must be completed by developers.
- Defects must be reported using a standardized tool (Jira).
- Test scripts require Test Lead approval before execution.

5 Test Approach

The project uses an approach with weekly iterations. Manual testing includes exploratory testing due to team unfamiliarity. Test cases use HTML element references for automation compatibility, covering functional, UI, and boundary scenarios comprehensively.

5.1 Test Levels

Level	Description
Level 1 – Build Acceptance	Basic tests (15–30 minutes) to verify application loads and accessibility.
Level 2 – Smoke Tests	High-level tests (20–30 minutes) for core functionalities.
Level 2a – Bug Regression	Tests to re-verify fixed bugs (5–60 minutes per bug).
Level 3 – Critical Path	Tests for daily-use features, executed per iteration.
Level 4 - Standard Tests	Comprehensive tests, run once per release cycle.

6 Test Environment

- **Browser**: Chrome (latest version)
- **Operating System**: Windows 10
- **Application**: To-Do List Web App (single-page)

- **Hardware**: Standard PC workstation
- **Software**: No additional software required

7 Milestones and Deliverables

7.1 Test Schedule

Task Name	Start	Finish	Effort	Comments
Test Planning	2025-08-01	2025-08-05	3 days	Define test cases
Review Requirements	2025-08-06	2025-08-07	2 days	RTM Validate with stakeholders
Create Test Estimates	2025-08-08	2025-08-08	1 day	Estimate test effort
Staff and Train Testers	2025-08-09	2025-08-10	2 days	Train on defect reporting
First Deploy to QA	2025-08-11	2025-08-11	1 day	Setup test environment
Functional Testing - Iteration 1	2025-08-12	2025-08-14	3 days	Execute functional tests
Iteration 2 Deploy to QA	2025-08-18	2025-08-18	1 day	New build deployment
Functional Testing - Iteration 2	2025-08-19	2025-08-21	3 days	Execute functional tests
System Testing	2025-08-22	2025-08-24	3 days	Execute UI, boundary tests
Regression Testing	2025-08-25	2025-08-26	2 days	Retest fixed bugs
UAT	2025-08-27	2025-08-28	2 days	End-user validation
Final Defect Resolution	2025-08-29	2025-08-30	2 days	Fix and verify bugs
Release to Production	2025-08-31	2025-08-31	1 day	Final deployment

7.2 Deliverables

Deliverable	For	Date/Milestone
Test Plan	Project Manager, Test Team	2025-08-05
Requirements Traceability Matrix	Project Manager, QA Lead	2025-08-05
Test Results Report	Project Manager	2025-08-30
Test Status Report	QA Lead, Project Manager	Weekly

8 Requirements Traceability Matrix

Req. ID	Requirement	Test Case ID
R1	Add task	FT-001, FT-002, FT-003, BT-001, BT-002, BT-003
R2	Edit task	FT-004, FT-005, BT-004
R3	Delete task	FT-006, FT-007
R4	Mark task complete	FT-008, FT-009
R5	View task list	FT-010, FT-011
R6	User login	FT-012, FT-013, FT-014
R7	UI usability	UI-001, UI-002, UI-003, UI-004, UI-005, UI-006
R8	Handle boundary cases	BT-001, BT-002, BT-003, BT-004, BT-005, BT-006

8 Automation Testing

8.1 Approach

Automation tests use Python and Selenium WebDriver to validate core functionalities, aligning with manual test cases (SM-001, SM-002, SM-003). The scripts are modular, reusable, and cover opening the application, adding a task, marking it as completed, and deleting it.